

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 05-054277

(43)Date of publication of application : 05.03.1993

(51)Int.Cl. G08B 23/00  
G06F 3/14  
G06F 15/72  
// G05B 23/02

(21)Application number : 03-211787

(71)Applicant : MITSUBISHI ELECTRIC CORP

(22)Date of filing : 23.08.1991

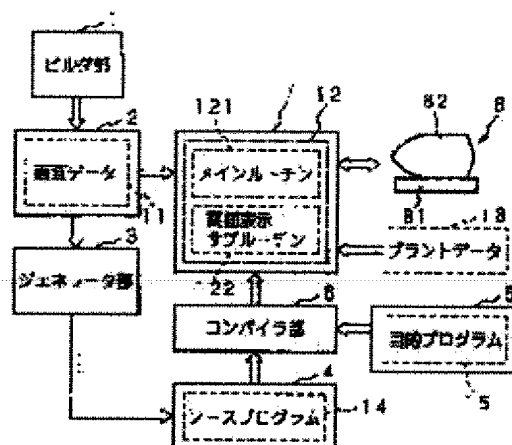
(72)Inventor : WAKE ASAOMI

## (54) PLANT MONITOR DEVICE

## (57)Abstract:

PURPOSE: To increase the processing speed regarding a display on a monitor screen.

CONSTITUTION: This plant monitor device is equipped with a means for storing an object program 15 equivalent to a main routine for performing processing regarding the state monitoring of a plant except screen display processing, a generator part 3 which generates a source program 14 equivalent to a subroutine, having screen data 11 in its program in advance and executing the screen display processing according to plant data 13, from the screen data 11, and a compiler part 6 which generates an execution module 12 consisting of screen display subroutines 122,... compiling the generated source program 14 and linking the compilation result with the object program 15 to perform screen display processing, and a main routine 121 including execution instructions of the screen display subroutines 122,... and performing processing regarding the state monitoring of the plant. The screen display processing is performed by the screen display subroutines 122, 122... of the generated execution module 12.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平5-54277

(43)公開日 平成5年(1993)3月5日

(51)Int.Cl. <sup>5</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 8 B 23/00	A	9177-5G		
G 0 6 F 3/14	3 2 0 C	8725-5B		
15/72	A	9192-5L		
// G 0 5 B 23/02	3 0 1 P	7208-3H		

審査請求 未請求 請求項の数1(全 9 頁)

(21)出願番号 特願平3-211787

(22)出願日 平成3年(1991)8月23日

(71)出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72)発明者 和氣 朝臣

兵庫県神戸市兵庫区和田崎町1丁目1番2

号 三菱電機株式会社制御製作所内

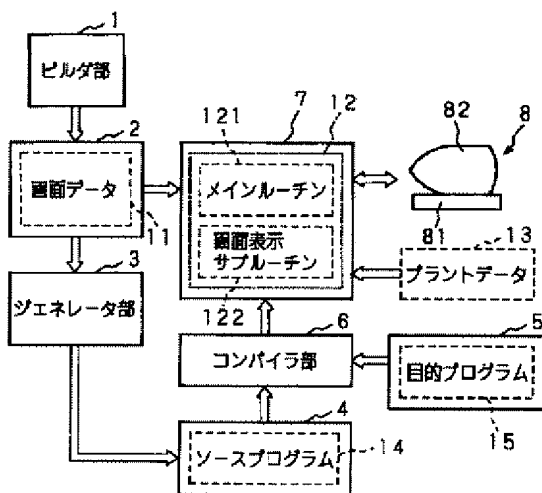
(74)代理人 弁理士 高田 守 (外1名)

(54)【発明の名称】 プラント監視装置

(57)【要約】

【目的】 監視画面の表示に関する処理速度の高速化を実現することを可能とする。

【構成】 画面表示処理以外のプラントの状態監視に関する処理を行うメインルーチンに相当する目的プログラム15を格納する手段と、画面データ11をそのプログラム中に予め組み込んであり、プラントデータ13に基づいて前記画面表示処理を実行するためのサブルーチンに相当するソースプログラム14を画面データ11から生成するジェネレータ部3と、生成されたソースプログラム14をコンパイルし、そのコンパイル結果と、目的プログラム15とをリンクすることにより、前記画面表示処理を行う画面表示サブルーチン122, 122 …と、該画面表示サブルーチン122, 122 …の実行指示を含む、プラントの状態監視に関する処理を行うメインルーチン121 とで構成される実行モジュール12を生成するコンパイラ部6とを備えており、生成された実行モジュール12の画面表示サブルーチン122, 122 …にて画面表示処理を実行する。



1…ビルダ部	11…画面データ
3…ジェネレータ部	12…実行モジュール
5…ライブラリファイル部	13…プラントデータ
6…コンパイラ部	14…ソースプログラム
7…実行部	15…目的プログラム
8…CRT端末部	121…メインルーチン
	122…画面表示サブルーチン

1

2

## 【特許請求の範囲】

【請求項1】 プラントの構成及びその稼働状態を監視画面として表示するために予め定めた画面データと、プラントの稼働状態の検出結果であるプラントデータとに基づいて、該プラントデータに応じた監視画面を表示させる処理である画面表示処理を含む、プラントの状態監視に関する処理を行う実行モジュールを実行させるようにしてあるプラント監視装置において、前記画面表示処理の実行指示を含む、プラントの状態監視に関する処理を行うメインルーチンを格納する手段と、前記画面データをそのプログラム中に予め組み込んであり、組み込んだ画面データと、前記プラントデータとに基づいて前記画面表示処理を行うサブルーチンを、前記画面データから生成する手段と、生成された前記サブルーチンと、格納された前記メインルーチンとに基づいて、前記メインルーチンが前記サブルーチンを実行させる構成とした実行モジュールを生成する手段とを具備することを特徴とするプラント監視装置。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】本発明は、プラントの構成及びその稼働状態を監視画面にて表示するプラント監視装置に関する。

## 【0002】

【従来の技術】プラントの構成及びその稼働状態を種々のシンボル及び数値にてCRT画面に表示し、この表示に基づいて、プラントの稼働状態の監視、制御を行うプラント監視装置が、種々のプラントにおいて用いられている。

【0003】図1は従来のプラント監視装置の構成を示すブロック図である。図中1は、プラントの構成及びその稼働状態を種々のシンボル及び数値にて画面表示するための画面データ11の作成及び編集を行うビルダ部である。このビルダ部1で作成された画面データ11は、画面データファイル部2に格納されるようになっている。画面データ11は、1画面に描画すべき機器の種類を表すデータ、画面上の任意の位置に任意の色でシンボルの描画を実行させる関数である描画関数、該描画関数の引数（前記位置、シンボルの形、色等を指定する引数）のデータ等の種々のデータにて構成されている。前記画面データファイル部2に格納された画面データ11は、後述する実行モジュール12に読み込まれるようになっている。

【0004】また、実行モジュールファイル部9は、プラントの構成及びその稼働状態を種々のシンボル及び数値にて表す監視画面を表示させるプログラムである実行モジュール12を記憶している。この実行モジュール12は、前記画面データ11の他に、図示しないプロセス出力装置からプラントの種々の機器の夫々の運転状態を表

すデータであるプラントデータ13を通信によって読み込むようになっており、読み込んだプラントデータ13と、前記画面データ11とに基づいて、後述する方法により、キーボードよりなる入力部81及びCRTよりなる表示部82を備えたCRT端末部8の前記表示部82に前記監視画面を表示させる処理を実行するようになっている。前記実行モジュール12は、プラント監視装置が起動されると、CPU及び主記憶装置よりなる実行部7にロードされ、実行されるようになっている。

【0005】次に以上の如く構成されたプラント監視装置の動作について説明する。まず、ビルダ部1にて、プラント監視に必要なプラント状態を表示するための画面データ11を予め作成し、それを画面データファイル部2に格納しておく。そして、プラント監視装置の起動時には、実行モジュール12が実行モジュールファイル部9から実行部7に読み込まれ、実行される。実行モジュール12は、まず、画面データファイル部2から画面データ11を読み込む。

【0006】そして、実行モジュール12は、CRT端末部8の入力部81から画面表示要求が与えられると、その画面の表示に必要なプラントデータ13を読み込み、読み込んだプラントデータ13と、予め読み込んだ画面データ11とを照合し、画面に表示すべき機器の状態を判別する。そして、この判別結果を参照し、画面に表示すべき夫々の機器を描画するための描画関数及びその引数を、読み込んだ画面データ11から夫々探し出し、画面に表示すべき全ての機器についての描画関数及びその引数を探し出した後、探し出した描画関数にその引数を設定し、この描画関数に基づいて監視画面をCRT端末部8の表示部82へ表示させる。

【0007】前記実行モジュール12による監視画面の表示方法を具体例で説明すると、例えば、ポンプが運転中であれば、緑色のポンプのシンボルを表示させ、ポンプが故障状態であれば、赤色のポンプのシンボルを表示させたい場合、プラントデータ13に基づいてポンプの運転状態を判別し、ポンプが運転中であると判別された場合は、緑色のポンプのシンボルを表示させる描画関数及びその引数を、画面データ11から探し出し、探し出した描画関数及びその引数に基づいて緑色のポンプのシンボルを描画した監視画面をCRT端末部8の表示部82へ表示させるということである。

## 【0008】

【発明が解決しようとする課題】しかしながら、前述の如き従来のプラント監視装置では、プラントデータ13に基づく各機器の状態の判別後に、夫々に対する描画関数及びその引数を探し出し、探し出した描画関数にその引数を設定する処理を行わなければならない、その処理においては、画面データ11へのアクセスが多数回必要であり、このため長い処理時間を要し、監視画面の表示に関する処理に長時間を要するという問題があった。

【0009】本発明はこのような問題点を解決するためになされたものであり、監視画面の表示に関する処理速度の高速化を実現することを可能とするプラント監視装置を提供することを特徴とする。

【0010】

【課題を解決するための手段】本発明に係るプラント監視装置は、画面表示処理の実行指示を含む、プラントの状態監視に関する処理を行うメインルーチンを格納する手段と、前記画面データをそのプログラム中に予め組み込んであり、組み込んだ画面データと、前記プラントデータとに基づいて前記画面表示処理を行うサブルーチンを、前記画面データから生成する手段と、生成された前記サブルーチンと、格納された前記メインルーチンとに基づいて、前記メインルーチンが前記サブルーチンを実行させる構成とした実行プログラムを生成する手段とを設けたものである。

【0011】

【作用】本発明にあつては、監視画面を表示させる画面表示処理を含む、プラントの状態監視に関する処理を行う実行モジュールが、前記画面表示処理を行うサブルーチンと、該サブルーチンの実行指示を含むプラントの状態監視に関する処理を行うメインルーチンとに分割構成されている。このため、前記実行モジュールでは、画面表示処理を前記サブルーチンが専門的に行うので、画面表示処理を含むプラントの状態監視に関する処理の負荷が分散される。前記サブルーチンには、画面データがそのプログラム中に予め組み込まれており、このサブルーチンでは、画面表示処理の際に、監視画面に関する画面データを読み込む必要がないため、プラントデータが得られると、そのプラントデータと、予め組み込んだ画面データとに基づいて、画面表示処理が高速で実行される。

【0012】

【実施例】以下本発明をその実施例を示す図面に基づいて具体的に説明する。図2は本発明に係るプラント監視装置の構成を示すブロック図である。

【0013】図中1はプラントの構成及びその稼働状態を種々のシンボル及び数値にて表す複数の監視画面を各別に表示するための画面データ11の作成及び編集を行うビルダ部である。このビルダ部1で作成された画面データ11は、画面データファイル部2に格納されるようになっている。画面データ11は、1画面に描画すべき機器の種類を表すデータ、画面上の任意の位置に任意の色でシンボルの描画を実行させる関数である描画関数、該描画関数の引数（前記位置、シンボルの形、色等を指定する引数）のデータ等の種々のデータより構成されている。前記画面データファイル部2に格納された画面データ11は、ジェネレータ部3及び後述する実行モジュール12における画面表示サブルーチン122, 122 に読み込まれるようになっている。

【0014】ジェネレータ部3は、画面データ11を入力として、後述するプラントデータ13から各種機器の運転状態等のプラントの稼働状態を判別し、その判別結果に基づいて各種機器の運転状態をシンボル及び数値にて表す監視画面を夫々1画面毎に表示させる処理を行うプログラムである複数の画面表示サブルーチン122, 122 ... を、ソースプログラム形式で出力するようになっている。

【0015】ジェネレータ部3から出力される画面表示サブルーチン122, 122 ... のソースプログラム14は、画面表示用ソースプログラムファイル部4に格納されるようになっている。また、プラント監視装置の基本動作を実行させる目的プログラム15及びマンマシンインターフェイス用の目的プログラム等のプログラムが、ライブラリとして実行用ライブラリファイル部5に格納されている。コンパイラ部6は、画面表示用ソースプログラムファイル部4に格納されたソースプログラム14のコンパイル及びそのコンパイル後のプログラムと、ライブラリファイル部5に格納された目的プログラム15とをリンクして、画面表示サブルーチン122, 122 と、プラント監視装置の基本動作及び画面表示サブルーチン122, 122 ... を実行させるメインルーチン121 とよりなる実行モジュール12を生成するようになっている。

【0016】コンパイラ部6によって生成された実行モジュール12は、プラント監視装置が起動されると、CPU及び主記憶装置よりなる実行部7にロードされ、実行されるようになっている。この実行モジュール12のメインルーチン121 によって実行される画面表示サブルーチン122, 122 ... は、図示しないプロセス入出力装置からプラントの種々の機器の夫々の運転状態を表すデータであるプラントデータ13を通信によって読み込むようになっており、このプラントデータ13に基づき前記機器の夫々の運転状態の判別を行い、その判別結果に応じた前記機器の運転状態を表す前記監視画面を、後述する方法によって、キーボードよりなる入力部81及びCRTよりなる表示部82を備えたCRT端末部8の前記表示部82に表示させる処理を実行するようになっている。

【0017】次に、以上の如く構成されたプラント監視装置の動作について説明する。まず、ビルダ部1にて、プラント監視に必要なプラント状態を表示するための画面データ11を予め作成し、それを画面データファイル部2に格納させておき、後述する作成手順により、ジェネレータ部3にて、前記画面データ11から画面表示サブルーチン122, 122 ... のソースプログラム14を生成し、該ソースプログラム14を画面表示用ソースプログラムファイル部4に格納しておく。

【0018】そして、プラント監視装置の起動時には、コンパイラ部6が、画面表示用ソースプログラムファイル部4に格納されたソースプログラム14のコンパイル及びそのコンパイル後のプログラムと、ライブラリファイ

5

ル部5に格納された目的プログラム15とのリンクを行って、画面表示サブルーチン122, 122 …と、メインルーチン121 とよりなる実行モジュール12を生成する。そして、生成された実行モジュール12は実行部7にロードされる。

【0019】ロードされた実行モジュール12のメインルーチン121 は、CRT 端末部8の入力部81から画面表示要求が与えられると、画面表示サブルーチン122, 122 …を実行させる。実行させられた画面表示サブルーチン122, 122 …は、その画面の表示に必要であるプラントデータ 13を読み込み、読み込んだプラントデータ13に基づいて、画面に表示すべき機器の状態判別を行う。そして、前記機器を前記状態判別の結果に応じたシンボルで描画する描画関数に基づいて、監視画面をCRT端末部8の表示部82へ表示させる。

【0020】次に、前記画面表示サブルーチン122 による画面表示処理の具体例について、説明する。例えば、ある監視画面において、表示すべき所定のポンプの形状を、円及び長方形を組み合わせたシンボルで表し、そのポンプが正常に運転している場合は前記シンボルを緑色、ポンプが停止している場合は前記シンボルを赤色、ポンプの回転数が低下している場合は前記シンボルを黄色で表すような画面表示サブルーチン122 があると仮定する。また、プラントデータ13のうちのii番目のデータ（以下プラントデータiiという）が前記ポンプの運転状態を表すデータであり、このデータiiの値は、ポンプが正常に運転している場合は“1”、ポンプが停止している場合は“2”、ポンプの回転数が低下している場合は“3”になると仮定する。

【0021】このようなポンプの運転状態を表示させる画面表示サブルーチン122 は、図3に示される如きソースプログラムとなる。図3は画面表示サブルーチン122, 122…のソースプログラムの一例を示すプログラムリスト図である。

【0022】図中21はサブルーチン宣言及びプラントデータ13の読み込み部分である前処理プログラムであり、前処理プログラム21の次は、表示すべきプラントデータiiを指定するプラントデータ指定文22となっている。

【0023】そして、プラントデータ指定文22の次は、ポンプが正常に運転している場合、即ち、プラントデータiiの値が“1”である場合に、ポンプを表す円及び長方形を組み合わせたシンボルを緑色で表示させる第1描画関数プログラム23、ポンプが停止している場合、即ち、プラントデータiiの値が“2”である場合に、前記シンボルを赤色で表示させる第2描画関数プログラム24、ポンプの回転数が低下している場合、即ち、プラントデータiiの値が“3”である場合に、前記シンボルを黄色で表示させる第3描画関数プログラム25、プラントデータiiの値がない場合に、プラントデータiiの値のデフォルト値として前記シンボルを緑色で表示させる第4

6

描画関数プログラム26が、この順に存在する。前記第1描画関数プログラム23～第4描画関数プログラム26の夫々には、夫々の関数を用いた描画が即実行可能なように引数が予め設定されている。

【0024】そして、第4描画関数プログラム26の次は、サブルーチンとしての形式を整えるための後処理プログラム27となっている。

【0025】次に、ジェネレータ部3におけるソースプログラム作成処理方法について説明する。図4はジェネレータ部3におけるソースプログラム作成処理手順の一例を示すフローチャートである。

【0026】まず、サブルーチン宣言及びプラントデータ13の読み込みを行う、前述の如き前処理プログラム21を生成する（ステップS1）。

【0027】次に、プラントデータ13の中から監視画面の1画面に表示すべきプラントデータ13を1つ抽出する（ステップS2）。そして、1画面中に表示すべき全てのプラントデータ13についての描画関数の生成が終了したか否かを判別する（ステップS3）。

【0028】ステップS3において、1画面中に表示すべき全てのプラントデータ13についての描画関数の生成が終了していないと判別された場合は、ステップS2で抽出されたプラントデータ13がとり得る状態の中の一状態を選択する（ステップS4）。そして、プラントデータ13がとり得る状態における全ての状態が既に選択されているか否かを判別する（ステップS5）。

【0029】ステップS5において、抽出されたプラントデータ13がとり得る状態における全ての状態が既に選択されていないと判別された場合は、ステップS4で選択された一状態に対応する全ての描画関数を収集し（ステップS6）、収集した描画関数に基づいて前述の如き描画関数プログラムを生成する（ステップS7）。前記ステップS4～ステップS7は、抽出されたプラントデータ13がとり得る全ての状態が選択されてしまうまで繰り返し実行され、これによって、例えば、前述の如き第1描画関数プログラム23～第3描画関数プログラム25を生成する。

【0030】また、ステップS5において、抽出されたプラントデータ13がとり得る状態における全ての状態が既に選択されていると判別された場合は、例えば、第4描画関数プログラム26の如きデフォルト状態の描画関数プログラムを生成する（ステップS8）。

【0031】また、前記ステップS3において、1画面中に表示すべき全てのプラントデータ13についての描画関数の生成が終了したと判別された場合は、前述の如き、サブルーチンとしての形式を整えるための後処理プログラム27を生成し（ステップS9）、処理を終了する。

【0032】このように、以上の如きプラント監視装置では、入力部81から監視画面の表示要求がなされた場合、画面表示サブルーチン122, 122 …のうち、前記表示要求に対応する画面表示サブルーチン122 が起動され、

起動された画面表示サブルーチン122が、その画面に必要であるプラントデータ13を読み込み、前述の如く監視画面を表示部82に表示させる処理を行う。このように監視画面の表示を実行する画面表示サブルーチン122では、描画関数に引数が予め設定されているので、プラントデータ13の状態を判別した後、直ちに監視画面の表示を行えるようになっており、このため、監視画面の表示を高速で実行する。

【0033】なお、本実施例では、実行用ライブラリファイル5に画面表示以外のプログラムのみを格納させていたが、これに限らず、表示すべき監視画面の全てに共通する処理に関するプログラムがある場合は、これを実行用ライブラリファイル5に、実行モジュール12によるアクセス可能に格納すれば、実行モジュール12のサイズを減らすことが可能である。

【0034】また、本実施例では、ジェネレータ部3は機械的にソースプログラムを出力するため、冗長なプログラムコードが生成される虞があるが、ジェネレータ部3に冗長なプログラムコードを検出させる機能を持たせれば、冗長なプログラムコードの生成を防ぐことが可能である。

【0035】また、本実施例では、実行モジュール12は1つだけ生成するようにしたが、これに限らず、実行モジュール12を1画面単位又は複数画面単位に分割しても良く、このようにすれば、実行モジュール12の処理速度を高速化することが可能である。

【0036】さらに、本実施例では、ジェネレータ部3にて生成されるソースプログラムは、プログラム言語によって構成されているので、表示すべき監視画面の総数が多いと、実行モジュール12のサイズが大きくなり過ぎて主記憶装置に乗らなくなる虞があるが、画面データ11の一部をソースプログラム化し、画面表示のための画面データを、画面データ11と、ソースプログラムとに分割して各別に格納させておき、これらの両者を画面表示のために用いるようにしても良く、このようにすれば、ソースプログラムのサイズを減少させることが可能であ

る。

【0037】

【発明の効果】以上詳述した如く本発明に係るプラント監視装置においては、実行モジュールが、画面表示処理を行うサブルーチンと、該サブルーチンの実行指示を含む、プラントの状態監視に関する処理を行うメインルーチンとに分割されており、画面表示処理を行うサブルーチンには、画面データがそのプログラム中に予め組み込まれているので、このサブルーチンでは、画面表示処理の実行の際に、監視画面に関する画面データを読み込む必要がないため、監視画面の表示に関する処理速度の高速化を実現することができる等、本発明は優れた効果を奏する。

【図面の簡単な説明】

【図1】従来のプラント監視装置の構成を示すブロック図である。

【図2】本発明に係るプラント監視装置の構成を示すブロック図である。

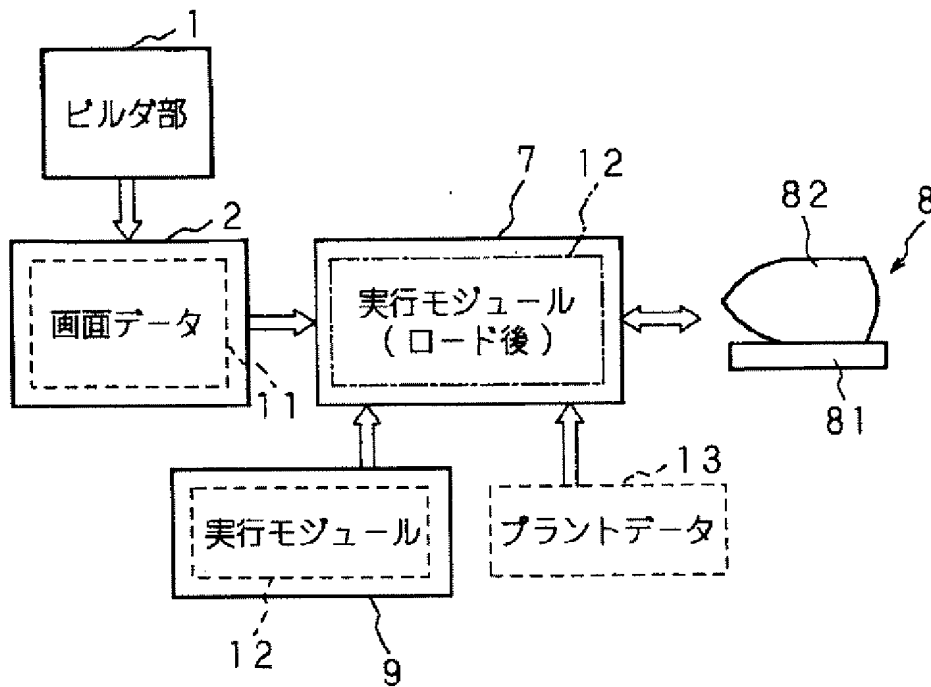
【図3】画面表示サブルーチンのソースプログラムの一例を示すプログラムリスト図である。

【図4】ジェネレータ部におけるソースプログラム作成処理手順の一例を示すフローチャートである。

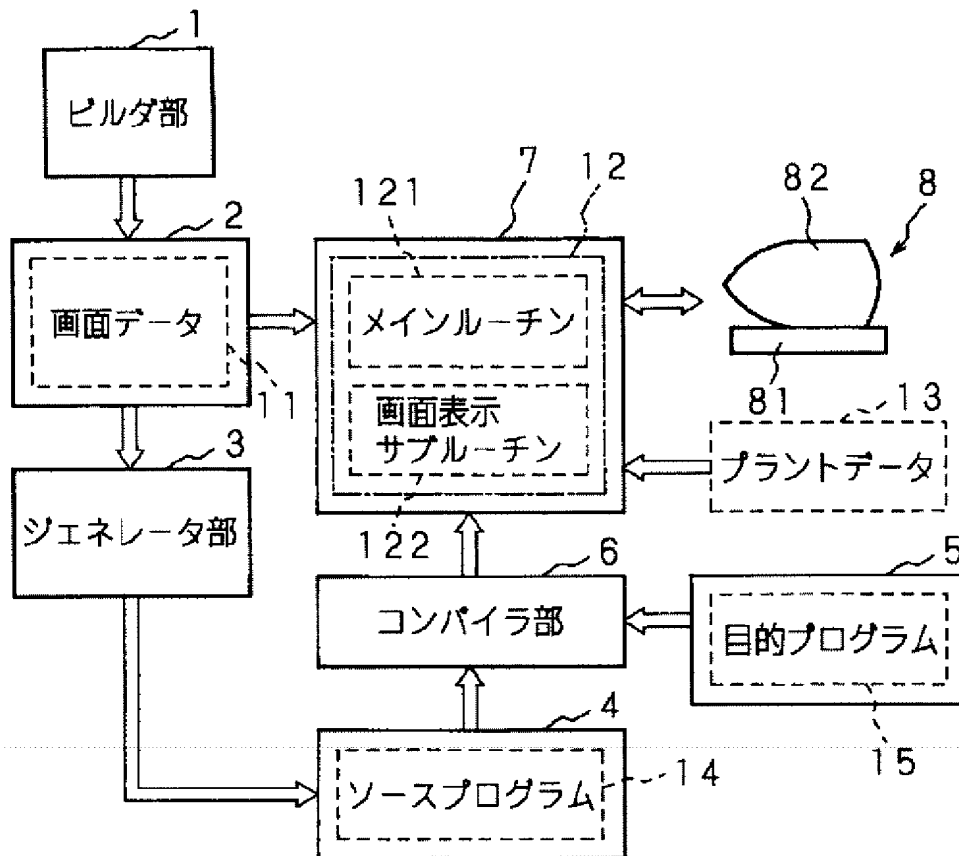
【符号の説明】

- 1 ビルダ部
- 3 ジェネレータ部
- 5 ライブラリファイル部
- 6 コンパイラ部
- 7 実行部
- 8 CRT 端末部
- 11 画面データ
- 12 実行モジュール
- 13 プラントデータ
- 14 ソースプログラム
- 15 目的プログラム
- 121 メインルーチン
- 122 画面表示サブルーチン

【図1】



【図2】



- |              |                |
|--------------|----------------|
| 1…ビルダ部       | 11…画面データ       |
| 3…ジェネレータ部    | 12…実行モジュール     |
| 5…ライブラリファイル部 | 13…プラントデータ     |
| 6…コンパイラ部     | 14…ソースプログラム    |
| 7…実行部        | 15…目的プログラム     |
| 8…CRT端末部     | 121…メインルーチン    |
|              | 122…画面表示サブルーチン |



【図3】

```

21 { subroutine gn1000
    .....
22 { switch(pdat[i])
    {
23 { case 1:
        call draw_circle(x,y,r,ct,cw,green)
        call draw_line(x1,y1,x2,y2,lt,lw,green)
    {
24 { case 2:
        call draw_circle(x,y,r,ct,cw,red)
        call draw_line(x1,y1,x2,y2,lt,lw,red)
    {
25 { case 3:
        call draw_circle(x,y,r,ct,cw,yellow)
        call draw_line(x1,y1,x2,y2,lt,lw,yellow)
    {
26 { default:
        call draw_circle(x,y,r,ct,cw,green)
        call draw_line(x1,y1,x2,y2,lt,lw,green)
    {
27 { end
        .....
        return
        end
    }
}
}

```

【図4】

